

## JASPA, la mejor alternativa libre a PostGIS

*J. Martinez-Llario<sup>(1)</sup>, M. Gonzalez <sup>(2)</sup> y E. Coll<sup>(3)</sup>*

<sup>(1)</sup> Departamento de Ingeniería Cartográfica, Geodesia y Fotogrametría, Escuela Técnica Superior de Ingeniería Cartográfica, Universidad Politécnica de Valencia, Camino de Vera s/n, 46022 Valencia, [jomarlla@cgf.upv.es](mailto:jomarlla@cgf.upv.es)

<sup>(2)</sup> Departamento de Ingeniería Cartográfica, Geodesia y Fotogrametría, Escuela Técnica Superior de Ingeniería Cartográfica, Universidad Politécnica de Valencia, Camino de Vera s/n, 46022 Valencia, [margonal@aaa.upv.es](mailto:margonal@aaa.upv.es)

<sup>(3)</sup> Departamento de Ingeniería Cartográfica, Geodesia y Fotogrametría, Escuela Técnica Superior de Ingeniería Cartográfica, Universidad Politécnica de Valencia, Camino de Vera s/n, 46022 Valencia, [ecoll@cgf.upv.es](mailto:ecoll@cgf.upv.es)

### RESUMEN

JASPA (Java Spatial) es una extensión espacial para bases de datos relacionales que dispongan de procedimientos almacenados en java. Se trata de un proyecto joven, pero que ya posee una funcionalidad muy similar a PostGIS. Recientemente han surgido nuevas extensiones espaciales, pero ninguna de ellas ofrece la misma funcionalidad que PostGIS o JASPA.

Uno de los principales objetivos de JASPA es establecerse como una sólida alternativa en java, lenguaje de programación ampliamente utilizado en el mundo del SIG libre, pero que actualmente no dispone de una extensión espacial de referencia para bases de datos.

Jaspa sigue un modelo simple features (OGC) y los estándares Simple Features For SQL y SQL/MM. En este momento ofrece alrededor de 200 funciones como predicados y operadores espaciales, arreglos de geometrías, agregados espaciales, etc. Se basa en las librerías de código abierto JTS y GeoTools.

Jaspa está programado en java y es sencillo de extender utilizando procedimientos almacenados en java. Actualmente da soporte a las bases de datos PostgreSQL y H2. Está prevista la portabilidad a HSQLDB y Oracle.

En este artículo se describe la arquitectura del producto, sus principales funcionalidades y las mejoras realizadas desde la última versión, entre las que destaca la inclusión de reglas topológicas.

**Palabras clave:** *Bases de datos espaciales, Java, HatBox, GeoDB*

## ABSTRACT

JASPA (Java Spatial) is a spatial extension for relational databases that provide java stored procedures. This is a young project, but it already has a very similar functionality to PostGIS. Recently many spatial extensions have arisen, but none of them offers the same functionality as PostGIS or JASPA.

One of the main goals of JASPA is to establish itself as a solid java alternative, a programming language widely used in the world of FOSS GIS, but it currently lacks of a spatial extension of reference for databases.

Jaspa follows the OGC simple features model and implements the OpenGIS Simple Features for SQL and the SQL/MM standard. It currently provides over 200 functions such as operators and predicates, spatial arrangements, spatial aggregates, etc. It is based on the open source libraries JTS and GeoTools.

Jaspa is programmed in Java and is easy to extend using java stored procedures. It currently supports PostgreSQL and H2 as backend Relational Data Base Management Systems. Portability to HSQLDB and Oracle is planned.

This paper describes the architecture of the product, its key features and improvements since the last version, most notably the inclusion of topological rules.

**Key words:** *Spatial Database, Java, HatBox, GeoDB*

## INTRODUCCIÓN

Los sistemas gestores de bases de datos se utilizan para manejar grandes cantidades de datos. Tradicionalmente, estos productos han sido diseñados para el manejo de datos de carácter alfanumérico, mientras que los datos de carácter geográficos han sido tratados en los SIG. El paradigma de la arquitectura integrada ha fomentado la centralización de toda la información en los SGBDs independientemente de su naturaleza.

De esta manera, se ha generalizado la utilización de extensiones espaciales, que proveen al SGBD de tipos de datos y funciones para manejar la información geográfica. Actualmente PostgreSQL y su extensión espacial PostGIS conforman la solución de código abierto más difundida entre los usuarios que requieren gestionar su información geográfica en una base de datos.

El software libre ha fomentado el desarrollo de muchos productos para el procesamiento de la información geográfica, ofreciendo a los usuarios una gama cada vez más amplia que cubre casi cualquier necesidad. Éste se suele clasificar en función de su utilidad (SIG de escritorio, servidores de mapas, etc.) o su lenguaje de desarrollo (principalmente C y Java). Cuando se describen las extensiones espaciales para bases de datos, tan solo aparecen opciones consolidadas en el lado del lenguaje C, como SQLite, MySQL y en especial PostGIS.

Por contra, los proyectos de extensiones espaciales programados en Java denotan menor madurez. En 2005 el equipo de GeoServer publicó SpatialDDBox y más tarde lanzó H2 spatial. Más recientemente Peter Yuli desarrolló el proyecto Hatbox y en 2010 Justin Deolivera publicó GeoDB. Si bien algunos de estos productos son una buena aproximación a PostGIS, presentan carencias como limitada funcionalidad o falta de índices espaciales que han dificultado su difusión.

## OBJETIVOS DEL NUEVO SOFTWARE

Dados estos antecedentes y recopilando las experiencias de trabajos con otras bases de datos, en especial PostgreSQL y PostGIS, se ha desarrollado una nueva extensión espacial programada en Java para satisfacer diversos objetivos:

- Crear una extensión espacial multiplataforma, que pueda utilizarse en diferentes SGBD, independientemente del lenguaje en el que este último se haya programado o su tipo de licencia.
- Completar el puzle del software libre para la información geográfica con una pieza que faltaba: una alternativa equiparable a PostGIS programada en Java. Al ser un lenguaje orientado a objetos potencia la reutilización del código, con menor tiempo de desarrollo y reduciendo la cantidad de código [1].
- Explotar las ventajas del lenguaje Java para el manejo de datos geográficos, especialmente en lo que respecta a la extensión del software por parte del usuario.
- Convertirse en un proyecto de investigación que pueda ser utilizado para la docencia, en el que los alumnos sean capaces de comprender como funciona una base de datos y desarrollar así sus propios algoritmos y funcionalidades.

Las ideas esbozadas en los párrafos anteriores se concretaron en un nuevo software denominado Java Spatial (abreviado como Jaspa) que se publicándose la primera versión 0.1 en verano de 2010 bajo licencia GNU GPL. En el siguiente apartado se describen sus principales características.

## CARACTERÍSTICAS GENERALES DE JASPA

Jaspa es una herramienta de software libre consistente en una extensión espacial para bases de datos relacionales desarrollada en lenguaje Java, proporcionando al SGBD en el que se despliegue operadores espaciales y predicados, colecciones de geometrías, etc. Puede ser instalado en sistemas operativos Windows o en Linux i386. Actualmente hay dos versiones de Jaspa dependiendo de la base de datos en la que puede ser desplegado:

- JASPA4PG para PostgreSQL
- JASPA4H2 para H2.

Jaspa implementa la especificación Simple Features Access del OGC [2]. Su evolución es el estándar internacional ISO 13249-3, también referido como SQL/MM Part 3 Spatial, que estandariza funciones para almacenar, recuperar y procesar la información geográfica mediante SQL [3], siendo parcialmente implementado por Jaspa.

Jaspa utiliza para el almacenamiento en la base de datos bloques binarios BLOB. Y por lo que respecta a la entrada de datos, ofrece soporte para los formatos Shapefile, KML y GML, WKT, WKB, EWKT y EWKB.

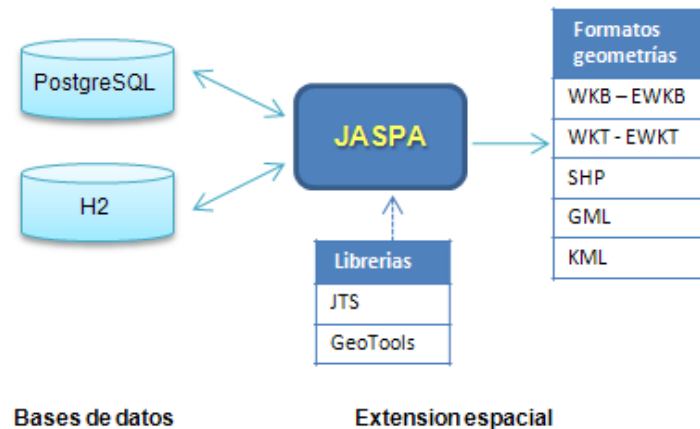


Figura 1: Arquitectura de la extensión espacial Jaspa

## FUNCIONES IMPLEMENTADAS

Las funciones de Jaspa están basadas en las librerías JTS (Java Topology Suite) [4] y GeoTools. La primera es una API (interfaz de programación de aplicaciones), especializada en predicados y funciones espaciales para un modelo geométrico basado en la especificación Simple Features Access del OGC, de amplio uso en el mundo del software java para la información geográfica. Por su parte, la librería GeoTools proporciona el uso de proyecciones cartográficas y el soporte de formatos de entradas como KML y Shapefile.

Jaspa proporciona actualmente 179 funciones para el manejo de la información geográfica que pueden ser clasificadas según su finalidad: gestión, relaciones espaciales, análisis, etc. (ver tabla 1). Las funciones están fuertemente basadas en las librerías JTS y GeoTools. Además, Jaspa implementa otras funciones que no están incluidas en estos estándares pero que son útiles para el trabajo con datos espaciales.

Además de incorporar funciones ya implementadas en las versiones 1.4 y 1.5 de PostGIS, en el proyecto Jaspa se están desarrollando nuevas funciones, entre las que destaca:

**ST\_Nodeline:** Dada una línea con auto intersecciones, este método parte la línea e inserta nodos en los puntos de intersección.

**ST\_CleanPolygon.** Este método fuerza a los polígonos a tener una geometría válida de acuerdo al estándar SFS.

**ST\_Snap:** Esta función intercepta los vértices y segmentos de una geometría dada con los de otra, mejorando la fiabilidad en operaciones de superposición mediante la eliminación de ejes paralelos muy cercanos que causan problemas durante cálculos de intersecciones. Se basa en la tolerancia cluster.

Tabla 1: Funcionalidad de Jaspa

Tipo de funciones	Ejemplos
Funciones de gestión	7 funciones (AddGeometryColumn, DropGeometryTable, Jaspa_Version ...)
Constructores de geometría	34 funciones (ST_GeomFromText , ST_MakeLine, ST_MakePolygon, ST_MakeGeomColl )
Acceso a geometrías	29 funciones (ST_CoordDim, ST_IsClosed, ST_IsValid ...)
Editores de geometría	27 funciones (ST_Affine, ST_Force_2D, ST_Multi, ST_NodeLine, ST_Transform ...)
Salida de geometrías	7 funciones (ST_AsBinary , ST_AsGML, ST_AsHEXEWKB, ST_AsText ...)
Relaciones espaciales y mediciones	30 funciones (ST_Centroid, ST_Contains, ST_Disjoint, ST_Distance, ST_Within ...)
Geoprocesamiento	19 funciones (ST_Buffer, ST_Collect, ST_Simplify, ST_Union ...)
Referencia lineal	8 funciones (ST_Line_Interpolate_Point, ST_Line_Substring, ST_Locate_Along_Measure)
Otras funciones	17 funciones (ST_Accum, ST_Box2D, ST_Expand, ST_Extent, ST_Xmax, ST_XMin ...)

### Creación de funciones por el usuario

La funcionalidad de Jaspa puede ser extendida utilizando el lenguaje procedural del SGBD en el que sea desplegado. Actualmente hay tres maneras de extender Jaspa:

- Utilizando procedimientos almacenados en PL/pgSQL en un sistema PostgreSQL-Jaspa.
- Utilizando procedimientos almacenados en PL/Java en sistemas PostgreSQL-Jaspa o H2-Jaspa. PL/Java es un módulo que posibilita la creación de procedimientos almacenamientos en Java para ser utilizados con PostgreSQL como base de datos [5].
- Utilizando clases nativas Java en sistemas PostgreSQL-Jaspa o H2-Jaspa. Las funciones creadas con este método tienen tiempos de ejecución más rápidos, debido a que utilizan geometrías JTS en lugar de BLOB, por lo que solo necesitan serializar y deserializar una vez.

## CONTROL DE LA TOPOLOGÍA

La topología aplicada a la información geográfica se utiliza principalmente para dos fines: por un lado, corrección geométrica de las entidades geográficas y por otro la ejecución de análisis espaciales. En el primer caso se dispone de información geográfica de estructura espagueti, siendo frecuente la presencia de polígonos superpuestos, nodos *dangle*, líneas con auto intersecciones, pseudonodos, etc., por lo que la depuración geométrica se utiliza para garantizar su calidad o incluso como paso previo para migrar la información a una estructura topológica [6].

Un método muy extendido en el mundo SIG para controlar la topología de información geográfica con estructura espagueti es la implementación de un conjunto de reglas topológicas que caracterizan el mundo real, previniendo incoherencias y garantizando consistencia espacial [7]. Se establecen relaciones y restricciones entre los elementos de una misma o distintas capas de información, permitiendo validar y detectar errores topológicos para su posterior corrección mediante procesos automáticos o manuales.

Con objeto de que el usuario pueda establecer un control de los errores topológicos desde la misma base de datos, la versión 0.2 de Jaspa incorpora un sistema de reglas topológicas similar al modelo de topología de gvSIG, Kosmo o ArcGIS. La particularidad de Jaspa es que es la primera extensión espacial para bases de datos de código abierto que incorpora una utilidad de topología de reglas. Así mismo añade un sistema de validación de reglas bajo demanda para que solo se validen aquellas regiones que potencialmente puedan haber cambiado.

### Fundamentos del sistema del modelo topológico de Jaspa

El cumplimiento de restricciones topológicas entre geometrías se basa en predicados binarios proporcionados por la librería JTS, que implementa el modelo DE-9IM (*Dimensionally Extended- 9 Intersection Model*) basado en la matriz de Eigenhofen [8]. Por tanto, la topología de en Jaspa se limita a estructuras bidimensionales.

Para mantener las relaciones topológicas entre los elementos es necesario utilizar una tolerancia cluster. Esta tolerancia cluster establece la distancia bajo la cual los vértices y los segmentos que conforman las geometrías son considerados idénticos. Se establece en función de la precisión utilizada para almacenar las coordenadas.

Jaspa implementa 63 reglas de topología, 28 de ellas para ser aplicadas en una capa y las otras 35 para ser aplicadas entre dos capas. Algunos ejemplos de reglas son que los polígonos no se superpongan, que una línea debe estar contenida en otra línea, o que un polígono debe contener un solo punto.

El proceso de control de topología en Jaspa podría resumirse en tres pasos:

- Creación del modelo de topología. Por cada modelo, se crean dos tablas de metadatos para almacenar las reglas que se utilizarán y las capas de información que intervendrán.
- Adición de capas al modelo y de las reglas que deben cumplir, especificando la tolerancia cluster para cada una de las capas.
- Validación de las reglas, que puede limitarse a una regla, a ciertas capas, o bien al modelo completo.

### Nuevas reglas topológicas

A diferencia de los sistemas existentes en los SIG de escritorio mencionados, en los que el conjunto de reglas está limitado, el usuario de Jaspa podrá extender el conjunto de reglas creando otras nuevas.

### ANÁLISIS COMPARATIVO

Dado que PostGIS es la extensión espacial de referencia en el mundo del software libre, cualquier producto que debe ser comparado con él para baremar sus características de forma rigurosa. En la tabla 2 se confrontan algunas características de Jaspa y PostGIS.

Tabla 2: Funcionalidad de Jaspa

Características	JASPA	PostGIS 1.4
Base de Datos	PostgreSQL y H2	PostgreSQL
Lenguaje de Programación	Java	C
Conectores SIG	PostgreSQL: Mapserver y Kosmo H2: -	QGis, gvSig, GeoTools, Gdal, MapServer...
Índices espaciales	PostgreSQL: Gist H2: -	Gist
Tratamiento de geometrías vacías	NULL	GEOMETRYCOLLECTION EMPTY
Cajas de geometría ( <i>bounding boxes</i> ) tratadas como geometrías OGC	Sí	No. Utiliza tipos box2d, box3d

Actualmente las extensiones espaciales son dependientes del SGBD al que proporcionan capacidades espaciales, por lo que Jaspa pretende ser un producto más versátil y universal que sea utilizable por usuarios independientemente del SGBD que utilicen para almacenar sus datos. Como se muestra en la tabla 2, Jaspa puede utilizarse con PostgreSQL y H2. En el primer caso se permite la creación de índices espaciales usando Gist de forma similar a PostGIS. Respecto a H2, compañeros de la la Universitat Oberta de Catalunya están desarrollando un índice espacial [9]. En un futuro Jaspa podría ofrecer también soporte en la base de datos java H2SQLDB y también en Oracle.

Una de las ventajas del software del código abierto es la posibilidad de que el usuario pueda manipular este código, adaptarlo, mejorarlo, etc. En este sentido, PostGIS puede ser difícil de extender debido a la utilización del lenguaje C, con el que muchos usuarios de SIG no están muy familiarizados, en su mayoría profesionales de ámbitos más o menos cercanos a las ciencias geográficas con un nivel de programación medio o bajo. Los procedimientos almacenados utilizando PL/pgSQL también son confusos, lo que unido al modo interno de manejar las geometrías de PostGIS con lwgeom impone demasiadas barreras para su extensibilidad. Con Jaspa lo que se pretende



posibilitar la extensión de la funcionalidad del producto mediante procedimientos almacenados en Java.

Así, Java se presentaba como la solución más adecuada, tanto por su facilidad de aprendizaje como por su amplia difusión. Desarrollar nuevos procedimientos almacenados a partir de las librerías JTS y GeoTools es más sencillo y ofrece más posibilidades que PostGIS con GEOS. También es posible la utilización de entornos de desarrollo integrados (IDEs) como Eclipse o NetBeans, que son más avanzados que los disponibles para PL/SQL y reducen el tiempo y el coste de desarrollo de software.

Uno de los puntos débiles de Jaspa por el momento es el acceso a los datos desde otros softwares. En la primera versión solo era posible la conexión mediante MapServer. No obstante, la versión 0.2 incorpora conectores para que los datos geográficos almacenados en PostgreSQL mediante Jaspa sean accesibles por gvSIG o Kosmo. La accesibilidad a los datos de H2 por otras aplicaciones es otro punto a desarrollar en próximas versiones.

Otro aspecto débil de Jaspa en comparación con PostGIS es que es un proyecto recién nacido. Uno de los principales problemas de los proyectos nuevos es la falta de una comunidad de usuarios estable en la que se produzca el feedback necesario para orientar el desarrollo del producto hacia las verdaderas necesidades de estos. Otro factor de fracaso de nuevos proyectos software es la falta de documentación del producto. En el caso de Jaspa, la publicación de nuevas versiones han venido acompañadas con la publicación de los correspondientes manuales descriptivos y tutoriales.

### **Ventajas del sistema de reglas topológicas**

El sistema de reglas topológicas proporcionado por Jaspa supone una novedad en el ámbito de las extensiones espaciales para SGBDs. Además, presenta diversas ventajas sobre sistemas parecidos implementados por los SIGs de escritorio actuales:

- Una base de datos puede albergar diversos esquemas de topología. Por ejemplo, una Feature Class de ArcGIS solo puede participar en una Geodatabase. Si se requiere que esa Feature Class participe en la topología de otra Geodatabase, se debe efectuar una copia de los datos. En Jaspa no se produce tal duplicación.
- A diferencia de ArcGIS, Jaspa permite establecer ciertas restricciones entre capas de multigeometrías.
- Las reglas pueden ser aplicadas a subselecciones de una capa de información que cumplan una cierta característica basada en su información alfanumérica asociada, y no sólo a toda la capa como sucede en Kosmo o gvSIG, y prescindiendo de la introducción de subtipos que requiere ArcGIS.
- En Jaspa no hay una única tolerancia cluster para todas las capas que participan en una topología. Esto es particularmente importante en conjuntos de datos en los que intervienen entidades geográficas que no han sido capturadas con un nivel de precisión equiparable.
- El usuario es capaz de definir sus propias reglas topológicas mediante programación en Java o SQL. De esta manera, cada usuario podrá obtener el



conjunto de reglas que mejor caracterice el comportamiento de sus datos.

- Existe control de errores, Jaspa no se limita a dar un error en tiempo de ejecución como ArcGIS. Si el error se produce por geometrías corruptas trata de corregirlas por funciones propias de Jaspa como ST\_CleanPolygon. Si el error no es automáticamente solventable devuelve una excepción con información precisa sobre la causa y las geometrías afectadas.

## CONCLUSIONES

Jaspa se encuentra en un momento de desarrollo. El futuro inmediato de Jaspa pasa por solucionar las carencias detectadas y difusión de las nuevas capacidades. En un año y medio de investigación se ha conseguido desarrollar un producto que emula la funcionalidad de PostGIS en lenguaje Java, implementado además nuevas funciones que no existen en ninguna otra extensión espacial de código abierto. Al estar programado en Java, puede incorporar las nuevas funcionalidades de la librería JTS de forma inmediata y se está trabajando para lograr la usabilidad de Jaspa en diferentes SGBD, siendo actualmente posible con PostgreSQL y H2.

La siguiente versión (Jaspa 0.2) incorpora conectores para Kosmo y gvSIG utilizando Jaspa en PostgreSQL y un conjunto de reglas para controlar la corrección topológica de los datos, siendo esta última una característica única en el ámbito de las extensiones espaciales para SGBDs. El sistema permite albergar dentro de una misma base de datos diferentes conjuntos de topología, reglas entre subselecciones de capas, asignación de diferentes tolerancias cluster o definición de reglas topológicas por el usuario. Es decir, el flujo de trabajo con esta herramienta será más versátil y flexible que el proporcionado en los sistemas de reglas de los SIGs de escritorio actuales.

Por lo tanto, Jaspa es un proyecto de software libre que puede ser de interés para usuarios de muy variados perfiles: aquéllos que utilicen normalmente PostgreSQL y que quieran probar una alternativa a PostGIS en Java, los usuarios de bases de datos ligeras como H2, usuarios avanzados que quieran extender la funcionalidad de Jaspa o la adapten a sus necesidades, etc. Precisamente, otra de las ventajas de ser software libre es la capacidad de mejora gracias a las aportaciones de usuarios y desarrolladores de software según estos detecten carencias en la usabilidad del producto.

## AGRADECIMIENTOS

Este trabajo ha sido parcialmente financiado por el proyecto de investigación "Creación y alimentación cartográfica de infraestructuras de datos espaciales en la administración local mediante un modelo de datos que integre catastro, planeamiento y patrimonio histórico" CSO2008-04808.

## REFERENCIAS

- ♦ [1] FICHMAN, R. G., & KEMERER, C. F. (2001). "Incentive compatibility and systematic software reuse". *[Electronic version]. Journal of Systems and Software*, 57(1), 45.
- ♦ [2] OPENGIS CONSORTIUM. (1999). "OpenGIS simple features specification for SQL, revision 1.1".
- ♦ [3] OpenGIS Consortium. (2006). "OpenGIS implementation specification for geographic information – simple feature access - part 2: SQL option" (revisión 1.2.1 ed.) de <http://www.opengeospatial.org/standards/sfs>
- ♦ [4] DAVIS, M. (2006). "JTS topology suite". Consultado el 10-15-2010, en <http://www.vividsolutions.com/jts/JTSHome.htm>
- ♦ [5] PL/JAVA. (2006). "PL/Java 1.2 user guide". Consultado el 11-14-2010 en [http://cvs.pgfoundry.org/cgi-bin/cvsweb.cgi/pljava/org.postgresql.pljava/docs/userguide.html?rev=1.15#Type\\_mapping\\_](http://cvs.pgfoundry.org/cgi-bin/cvsweb.cgi/pljava/org.postgresql.pljava/docs/userguide.html?rev=1.15#Type_mapping_)
- ♦ [6] BAARS, M., STOTER, J.E., VAN OOSTEROM, P.J.M. AND VERBREE, E. (2004) "Rule - based or explicit storage of topology structure: a comparison case study." *AGILE 2004 : proceedings of the 7th conference on geographic information science*, 9 abril - 1 mayo, 2004. Heraklion, Greece. Crete University Press, 2004. pp. 765-769.
- ♦ [7] BROWN, D. G., RIOLO, R., ROBINSON, D. T., NORTH, M., & RAND, W. (2005). "Spatial process and data models: Toward integration of agent-based models and GIS". *[Electronic version]. Journal of Geographical Systems*, 7(1), 25-47.
- ♦ [8] C. STROBL. (2008) "Dimensionally Extended Nine-Intersection Model (DE-9IM)," *Encyclopedia of GIS*, S. Shekhar and H. Xiong, Eds., Springer, 2008, pp. 240-245.
- ♦ [9] J.A. CALVILLO, J.M. DE DIEGO, A. PÉREZ-NAVARRO (2010) "Desarrollo de un índice espacial para la extensión JASPA sobre la base de datos H2". *Universitat Oberta de Catalunya*.